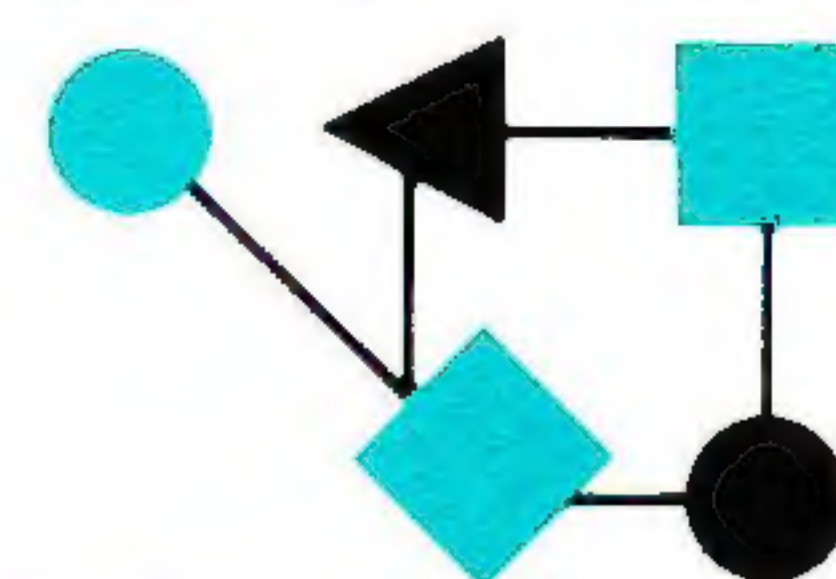


CONNEXIONS



The Interoperability Report

October 1995

Volume 9, No. 10

ConneXions —
The Interoperability Report
tracks current and emerging
standards and technologies
within the computer and
communications industry.

In this issue:

IPv6 Transition.....	2
Virtual Collaboration.....	18
Announcements.....	26
Book Reviews.....	28
Letters to the Editor.....	30

ConneXions is published monthly by Interop Company, a division of SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California, 94404-1138, USA.

Phone: +1 (415) 578-6900

Fax: +1 (415) 525-0194

E-mail: connexions@interop.com

Subscription hotline: 1-800-575-5717
or +1 610-892-1959

Copyright © 1995 by Interop Company.
Quotation with attribution encouraged.

ConneXions—The Interoperability Report
and the *ConneXions* logo are registered
trademarks of Interop Company.

ISSN 0894-5926

From the Editor

"The Internet is about to become a victim of its own success." So begins a just-published book on the *Internet Protocol Next Generation* (IPng), also known as IP Version 6 (IPv6). As you already know, the Internet has grown rapidly in recent years and there is concern in the engineering community that we might soon run out of IP addresses from the current 32-bit address space. The IPng effort expands the addressing space to 128 bits, enough to last us a *long* time. The textbook is a collection of papers by many prominent members of the Internet community. We have obtained permission to publish several extracts from this important book and we begin this month with a look at how one might go about transitioning from an existing IP Version 4 system to IPv6. Ease of transition was one of the major design goals for IPv6, but many users still wonder exactly how and when they will start using IPv6. The article is by Bob Gilligan and Ross Callon.

Virtual Reality (VR) has become more than just a buzzword. Several applications of VR are under development both as stand-alone entities or distributed networked systems. Steve Benford and Chris Greenhalgh describe *Collaborative Virtual Environments* (CVEs), distributed multi-participant virtual reality systems that are being developed on the Internet. I encourage you to take a look at some of the screenshots available from their World-Wide Web site. This is one time when I wish we used more than two colors in our publication.

Speaking of printing technology, if you compare this month's issue to any previous edition you will notice a subtle change in color. This is due to a change in paper. From now on we will be using Skyland Opaque Gray from the Champion International Paper Company in place of the (discontinued) Gray Mustang from Simpson. According to Champion's flyer, this paper is made from "elemental-chlorine-free pulp" and contains 20% recovered fiber, all of which is post-consumer.

Multimedia is another topic which has gone beyond the buzzword stage. But proper understanding of any emerging technology depends on well-written books that explain the state of the art. We are happy to bring you reviews of two such books on multimedia. The reviews are written by multimedia guru Jon Crowcroft of University College London.

We don't often receive feedback from our readers regarding particular articles, but two recent essays on "Protocol Wars" and "Spamming" gave rise to several letters. As always we appreciate you input, keep those letters coming to: connexions@interop.com

IPv6 Transition Mechanisms Overview

by
Robert E. Gilligan, Sun Microsystems, Inc.
and
Ross Callon, Bay Networks, Inc.

Introduction

The IETF has invested considerable energy in the design of transition mechanisms that will ensure a straight-forward, graceful evolution from IPv4 to IPv6. If not managed carefully, the cost, complexity, and hassle of transitioning to IPv6 could easily deter users from upgrading to the new protocol. To avoid these pitfalls, the transition mechanisms are intended to make the adoption of IPv6 as unintrusive as possible for both end users and administrators, allowing the benefits of the new IPv6 features to become a motivating factor in IPv6 adoption while alleviating potential downsides.

One of the transition mechanism designers' primary goals was to allow as much flexibility as reasonably possible. Because different user communities will have different transition requirements, a wide variety of transition scenarios are supported, as well as the capability for measured, incremental deployment. If sites were required to upgrade a group of machines in concert (e.g., they must upgrade all machines on a subnet at the same time), the difficulty of coordinating the work might discourage them. Instead, users should be able to deploy IPv6 one host or router at a time.

As much as possible, the transition strategy avoids dependencies between the various elements of a network during the upgrade process. If a user must wait for some other machine to be upgraded to IPv6 before his or her own machine can be upgraded (e.g., if routers must be upgraded before hosts can be), the transition may be delayed. To combat this difficulty, IPv6 can be introduced in hosts first, in routers first, or in both, on an as-needed basis. Wherever possible, the transition model creates no prerequisites for upgrading any specific host or router to IPv6 before the upgrade process can proceed.

Existing IPv4 infrastructures have been built up over a long period of time and at great expense. Consequently, users will not recklessly discard these assets in order to deploy IPv6. [10] To leverage the installed base and minimize start-up costs, the transition strategy lets IPv6 nodes exploit in-place IPv4 resources to whatever degree is appropriate for each site.

To accommodate the full range of potential IPv6 adopters, the IPv6 transition mechanisms include two major elements that work independently or in a synergistic fashion:

- *Dual-IP layers in hosts and routers:* Name servers and routers provide support for both IPv4 and IPv6 throughout the transition period. Hosts are gradually upgraded over a period of time. This allows upgraded nodes to interoperate with both IPv4 and IPv6 nodes using their native protocol.
- *Tunneling IPv6 over IPv4:* Hosts (and optionally routers) can tunnel IPv6 traffic through IPv4 routing topologies by encapsulation. This capability leverages the existing installed IPv4 routing system and allows IPv6 operation to get started early.

With these two mechanisms, network designers have the freedom to decide for themselves how transition is achieved. Some sites will upgrade a number of hosts first and take advantage of IPv6's ability to tunnel through an IPv4 routing fabric.

Other sites will upgrade some or all routers first to dual-IP layers, laying a foundation for any IPv6 hosts that come on line as transition proceeds. Still other sites may use a combination of these approaches.

In most cases, pockets of an enterprise will convert to IPv6 first, while other areas will remain with IPv4 for an extended period of time. With the tunneling and dual-IP features, IPv6 does not require that all the nodes in an IP area or even a subnet be converted all at once, so some sub-areas may contain both protocols.

Beyond the basic mechanics of dual-IP nodes and IPv6 tunneling, the IETF transition work addresses the implications of routing in highly heterogeneous IPv4/IPv6 networks, encompassing the various ways that topology structures and routing protocols (e.g., OSPF, RIP) operate in mixed environments. This article contains basic overviews of dual-IP-layer nodes and tunneling, and concludes with a look at routing operation in the presence of tunnels.

Dual-IP Layer networks

One of the important features of the IPng transition is that it allows IPv6 support to be added to the network over a period of time, without disrupting the existing IPv4 infrastructure. This is accomplished via a “dual-IP layer” transition scheme that allows IPv6 to be added to hosts, DNS servers, and routers, without any change or disruption in the existing IPv4 support.

The IPng transition therefore makes use of a simple long-term transition model that allows gradual update of Internet Hosts to run internet applications over IPv6, while, in parallel, name-to-address lookup resources (such as DNS) are updated so they are able to return IPv6 addresses. The basic model outlined in this section assumes that routers will be updated to support forwarding of IPv6 (in addition to IPv4) before IPv6 is introduced into hosts. However, the following section describes tunneling techniques which can be used to eliminate the dependency on early router updates.

The basic dual-IP layer transition method allows existing Internet transport and application protocols to continue to operate unchanged, except for the replacement of 32-bit IPv4 addresses with 128-bit IPv6 addresses. This will have some effect on applications and associated APIs.

In a dual-IP layer transition, hosts are updated to be able to use either IPv4 or IPv6 depending upon the capability of the corresponding host that they are contacting. Major servers and other hosts can run dual-IP layers indefinitely (until they are converted solely to IPv6, after completion of an extended transition period). Hosts with dual-IP layers can talk directly to IPv4 or IPv6 partners and they can use either IPv6 or IPv4 routes to communicate.

Figure 1 illustrates the basic operation of a dual-IP layer (IPv4 and IPv6) network. The figure shows a single Internet Routing Domain, which is also interconnected to Internet backbones and/or regionals. Nodes include two dual-IP layer hosts, N1 and N2, as well as two IPv4-only hosts H1 and H2, plus a *Domain Name System* (DNS) server and two border routers. It is assumed that the routers internal to the routing domain are capable of forwarding both IPv4 and IPv6 traffic. Typically this would be accomplished by using multi-protocol routers which can forward both protocol suites (although it would be possible to use a different set of routers for each suite).

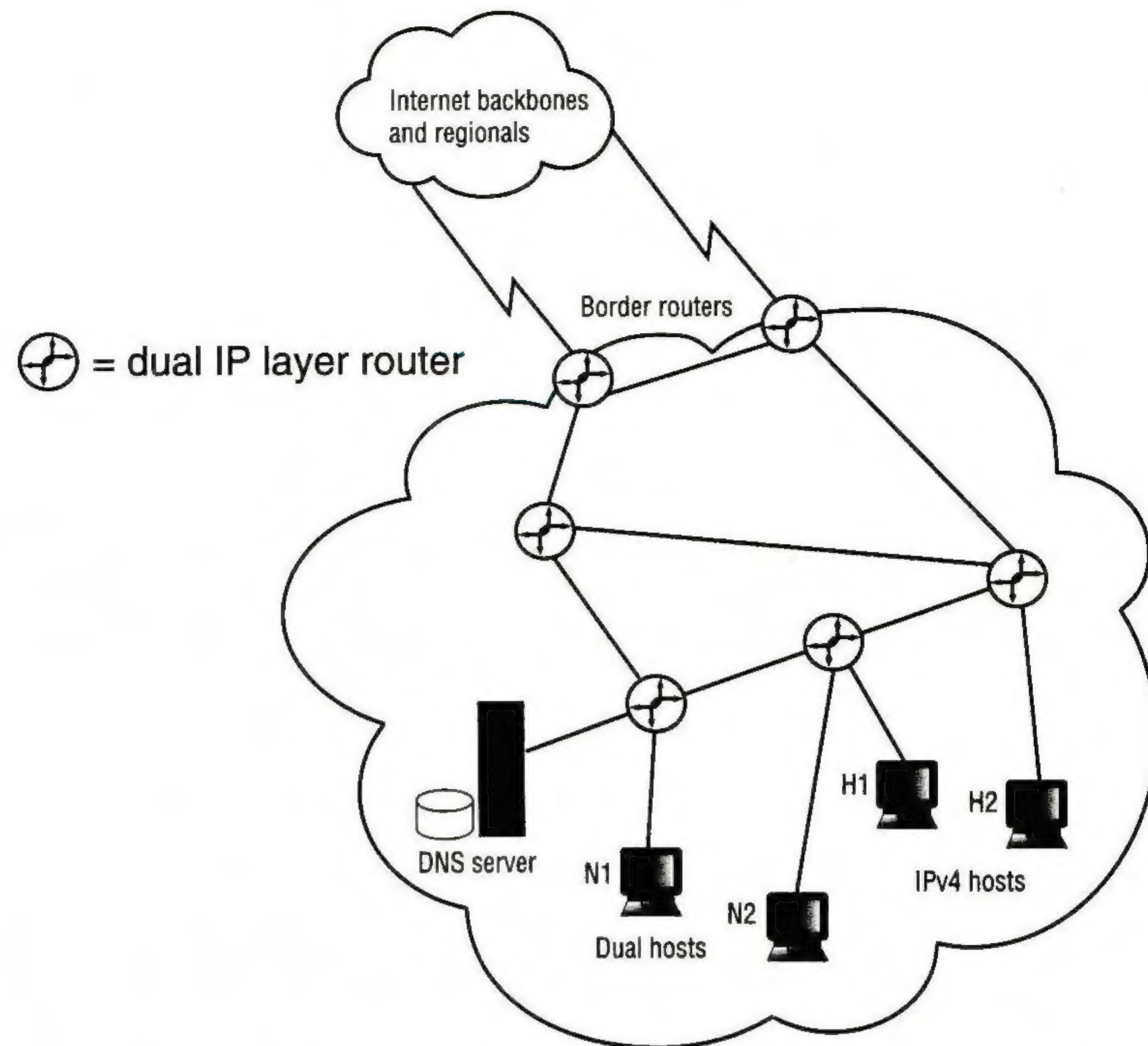
IPv6 Transition Mechanisms (*continued*)

Figure 1: Overview of a Dual-IP network

Dual hosts talk to IPv4 hosts using IPv4 unchanged. Dual hosts talk to other dual hosts using IPv6. This implies that dual hosts are able to send and receive either traditional packets (using IPv4), or new style packets (using IPv6). Which type of packet to send is determined via the normal name-to-address lookup. In the IPv6 and dual-IP layer environments, DNS servers are upgraded to accommodate a new record type that handles the 128-bit addresses of IPv6. In some cases, this function will alternatively be provided by 128-bit local host tables.

Suppose that host N1 wants to communicate with host H1. N1 asks its local DNS server for the address associated with H1. In this case, since H1 is not updated, the address available for H1 is an IPv4 address, and thus the DNS response returned to N1 specifies a conventional 32-bit IP address, letting N1 know that it needs to send an IPv4 packet to H1.

Now suppose that host N1 wants to communicate with host N2. Again, N1 contacts the DNS server. If the DNS record type for N2 has not been updated, then the DNS server will respond with an IPv4 address, and the communication between N1 and N2 will use IPv4. In this case, however, assuming that the DNS server has been updated to be able to return 128-bit addresses (and that the appropriate resource records have been configured into the DNS server), the DNS server will respond to N1 with the IPv6 address for N2. This allows N1 to know to use IPv6 instead of IPv4 for communication with N2 (more on the DNS below).

Addressing in a Dual-IP Layer network

A major aspect of the IPv6 transition plan is the assignment of IPv6 addresses to hosts and routers. As in traditional IPv4 environments, IPv6 routers will typically receive address assignment manually as part of the topology definition process. Address assignments for IPv6 hosts can be manual or can take advantage of the full range of automatic assignment mechanisms which are being defined for IPv6, including the new stateless autoconfiguration services.

Dual IPv6/IPv4 nodes need to be configured with both IPv4 and IPv6 addresses. Although the two addresses may be related to each other, this is not required. It is permissible for IPv4 and IPv6 address assignments to be completely independent. For example, there are some environments in which IPv4 hosts have been assigned local addresses (i.e., addresses which are not globally unique). This is typically done in situations where the IPv4 host does not require global Internet access, or where sufficient addresses were not available.

If it were later determined that a host with an IPv4 address required global Internet access using IPv6, then it would be perfectly reasonable to assign the host a global IPv6 address even though its IPv4 address is local. Note that the use of globally significant addresses may be desirable even for systems which do not require Internet access for two reasons: one, because it eliminates one possible source of confusion in network management; and two, because it allows consistent addressing based on a single prefix for all systems on a network, while simultaneously allowing for the possibility that some of these systems may require Internet access.

For sites that require global Internet access, it is desirable to assign IPv6 addresses in a manner that facilitates global Internet routing. The best way to do this is a topic of current research and debate. At the time of writing, the best-known method for address assignment that facilitates routing makes use of topologically significant addresses based on provider administration of the addresses. This has become known as "CIDR" (*Classless Inter-Domain Routing*). [9]

Updating the DNS

The Domain Name System (DNS) is used in both IPv4 and IPv6 to map host names into IP addresses. A new DNS resource record type named "AAAA" has been defined for IPv6 addresses. The IPv4 DNS records are referred to as type "A," (for address record). IPv6 records are AAAA or "quad A" because IPv6 addresses are four times larger than the 32-bit IPv4 addresses. Since dual-IP layer hosts need to be able to resolve host names (e.g., `ds.internic.net`) into either IPv4 or IPv6 addresses, they must be capable of dealing with both record types.

Before a DNS server can service IPv6 and dual-IP layer clients, it must be upgraded to handle the new AAAA record type. DNS servers are typically hard-coded for certain record types, so AAAA support will be provided by system software vendors and other TCP/IP utility sources. Fortunately, DNS software may well be one of the first things that vendors will offer because IPv6 support is a relatively easy change.

Note that, since the queries for IPv6 addresses from dual-IP layer clients can utilize IPv4, the DNS servers that provide AAAA record support need not necessarily themselves be upgraded to make use of IPv6 for data transfer between DNS servers.

For sites that have not yet upgraded their DNS, IPv6 nodes may resolve network names to addresses by using manually defined local host tables. These are files that reside on hosts that map names to IP addresses. Use of host tables may be particularly useful in the very early stages of transition before the DNS infrastructure has been converted to support AAAA records. The local host table mechanism does not scale very well, however, so its use is not recommended for large sites. This is because local tables only work if the network administrator manually enters the names and addresses of all the hosts that the host needs to communicate with.

IPv6 Transition Mechanisms (*continued*)

Application issues

With the dual-IP layer approach to transition, traditional IPv4 applications can run indefinitely over the IPv4 routing infrastructure without any modifications whatsoever. New applications can be written to IPv6 exclusively, or to both protocols, with minimal overhead. Although IPv4-based applications can continue to operate for the foreseeable future, at some point, as IPv6 takes hold and IPv4 addresses run out, it is likely that IPv4 applications will be limited to a scope that is local within individual companies (that is, IPv4 may continue to be suitable for certain applications that typically do not require global connectivity, such as printing).

In the IPv6 transition process, no enhancements need be made to non-network applications or to applications that indirectly access the network via resident network services such as FTP, SMTP, Telnet, etc. Applications that directly access the IPv6 stack will of course have to be upgraded before they can use the network. This includes FTP, SMTP, *telnet*, and *rlogin* utilities, as well as network-aware database and office automation applications.

At a minimum, network applications will have to be enhanced to request AAAA records from the DNS and to pass the 128-bit addresses to the local TCP/IP socket or similar interface. An upgrade to handle the new address space will require as little as a few lines of code change. Applications that take advantage of IPv6's advanced features (security, flow control, encryption, etc.) will require more extensive changes.

An extension to the standard TCP/IP sockets interface has been written and introduced into the public domain. This specification helps developers write applications that use TCP and UDP over IPv6. [6]

Routing and Dual-IP Layer networks

The dual-IP layer transition allows IPv4 and IPv6 to be routed independently. Much of the flexibility of the IPv6 transition strategy stems from the fact that routers already deal with multiple protocols. It is common in today's networks to find routers that support IPv4, IPX, DECnet, SNA, AppleTalk, and other protocols simultaneously. Usually each protocol is supported by a separate routing protocol using independent addressing structures. Hence, an additional protocol (IPv6) is not a major change.

The addition of IPv6 for routers can be somewhat simplified (relative to the addition of other protocol suites) in that IPv6 can use the same routing protocol (RIP or OSPF) as IPv4. Similarly, IPv6 can be managed using the same management protocol (SNMP), as well as using the same name service (DNS). Thus, there is a considerable administrative overlap with the in-place IPv4 infrastructure, and the "learning curve" required to add IPv6 support is minimized.

One possible minor enhancement entails the use of a single instance of an integrated routing protocol to support routing for both IPv4 and IPv6. At the time of writing there is no protocol that has yet been enhanced to support this feature. But if such a protocol is developed, it would not change the basic dual-IP layer nature of the transition.

When completely independent routing functions are employed, forwarding of IPv4 packets is based on routes learned through running IPv4-specific routing protocols, while forwarding of IPv6 packets is based on routes learned through running IPv6-specific routing protocols.

Structuring Dual-IP networks

This structure implies that separate instances of routing protocols are used for IPv4 and for IPv6, although it could consist of two instances of OSPF and/or two instances of RIP (since both OSPF and RIP are capable of supporting both IPv4 and IPv6 routing).

With a pure dual-stack transition strategy, the architectures of the IPv6 and IPv4 networks can be logically decoupled, even though they are running on the same physical infrastructure. In this case, existing IPv4 functions remain independent and unaffected, including such aspects as the feeding of route information between routing domains and local address assignments. In effect, the IPv6 topology is built from scratch.

Although some network designers may want to think of the new IPv6 architecture as entirely independent, there are some advantages to aligning the structures of the two logical networks. In many cases the IPv4 and IPv6 domain boundaries may be the same, so the enterprise's backbone and organizational structure can be retained. This involves using the same domain boundaries and the same area boundaries for partitioning the topology.

There are many efficiencies to this approach, but as long as the new IPv6 network uses true dual-IP layer and true IPv6 addresses, it can be considered an independent architecture that can evolve in its own direction. If for instance, the current IPv4 addressing is not ideal (e.g., designers fail to use CIDR-based addressing) or if IPv4 has an awkward area structure, these problems could be fixed in an independent IPv6 network with minimal effort.

Tunneling

In most deployment scenarios, the IPv6 routing infrastructure will be built up over time. Tunneling provides a way to use an existing IPv4 routing infrastructure to carry IPv6 traffic for as long as there are no native IPv6 resources to exploit. While IPv6 is being deployed, the existing IPv4 routing infrastructure can remain functional and can be used to carry IPv6 traffic. IPv6/IPv4 hosts and routers tunnel IPv6 datagrams over regions of IPv4 routing topology by encapsulating them within IPv4 packets. Tunneling can simplify the transition process for users as well as providing a number of other advantages:

Tunneling leverages the existing IPv4 routing system to build the IPv6 routing system. In cases where the initial IPv6 topology represents a small portion of the total network, it may be significantly more efficient to tunnel IPv6, rather than build a completely new IPv6 topology.

Tunneling helps activate a global IPv6 service early on in the transition. Early adopters of IPv6, who may be geographically dispersed, can use tunneling to provide global IPv6 connectivity without waiting for the entire Internet to be converted to IPv6.

Tunneling supports a transition strategy in which the IPv6 routing infrastructure can be built incrementally over time. It allows the IPv6 infrastructure to be brought to the sites where it is needed when it is needed.

Although there are a variety of tunneling methods, most of the underlying mechanisms are the same. To send a packet into a tunnel, a node first creates and prepends an encapsulating IPv4 header, and then transmits the encapsulated packet, as shown in Figure 2.

IPv6 Transition Mechanisms (*continued*)

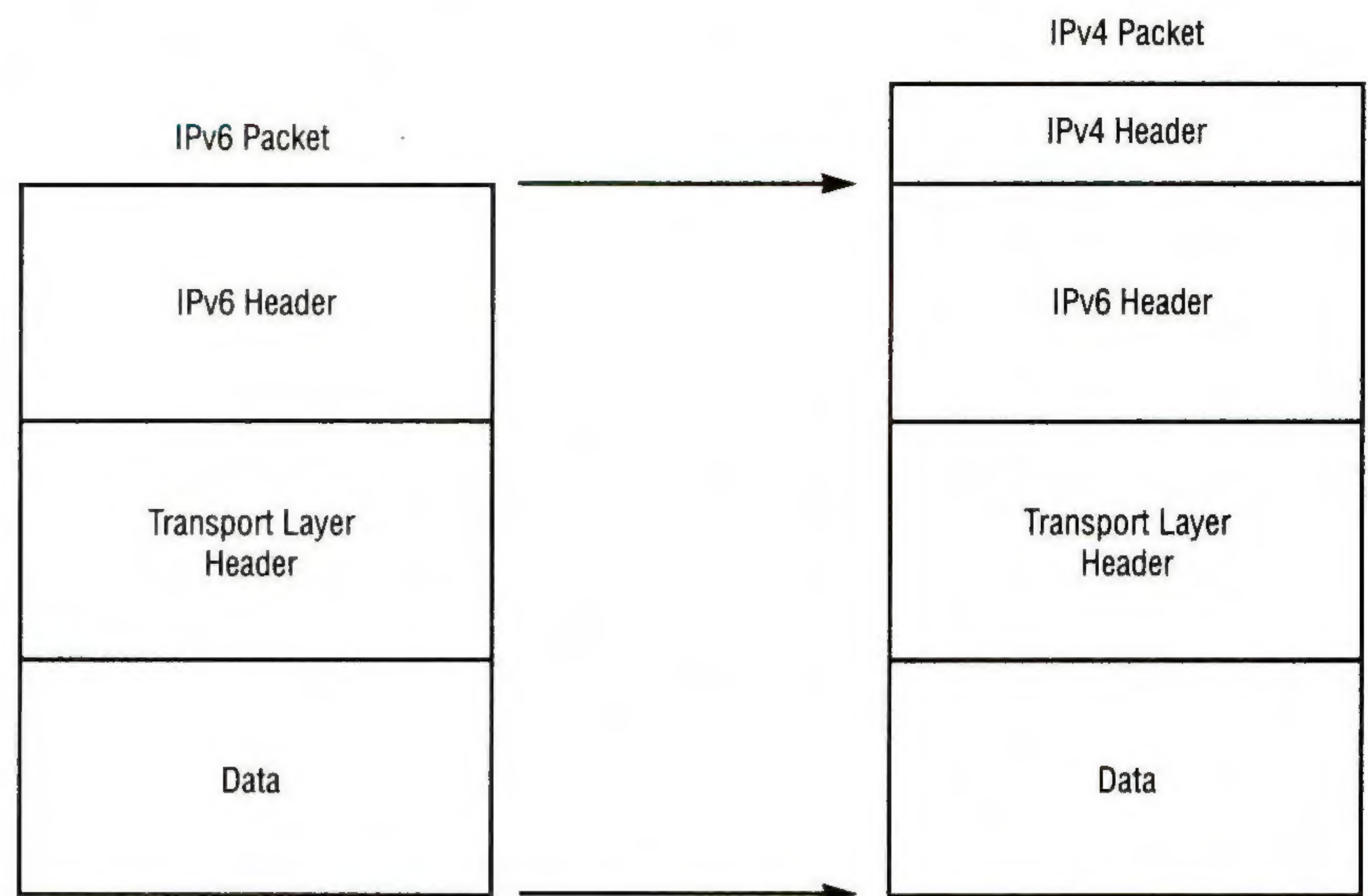


Figure 2: Encapsulating IPv6 in IPv4

The destination address of the encapsulating IPv4 packet specifies the tunnel endpoint—the node that receives the encapsulated packet strips off the encapsulating IPv4 header, updates the IPv6 header, and then processes the enclosed IPv6 packet as it would any other received packet.

Automatic versus configured tunneling

Two major tunneling methods are available: configured tunneling and automatic tunneling. Configured tunneling employs traditional tunneling methods in which individual logical “tunnel links” are configured between two nodes, typically routers, that are separated by an arbitrary IPv4 topology. These logical, layer-3 links are treated by tunneling nodes as virtual point-to-point connections. Each tunnel link is configured manually by assigning IP addresses to one or both tunnel endpoints (see Figure 3).

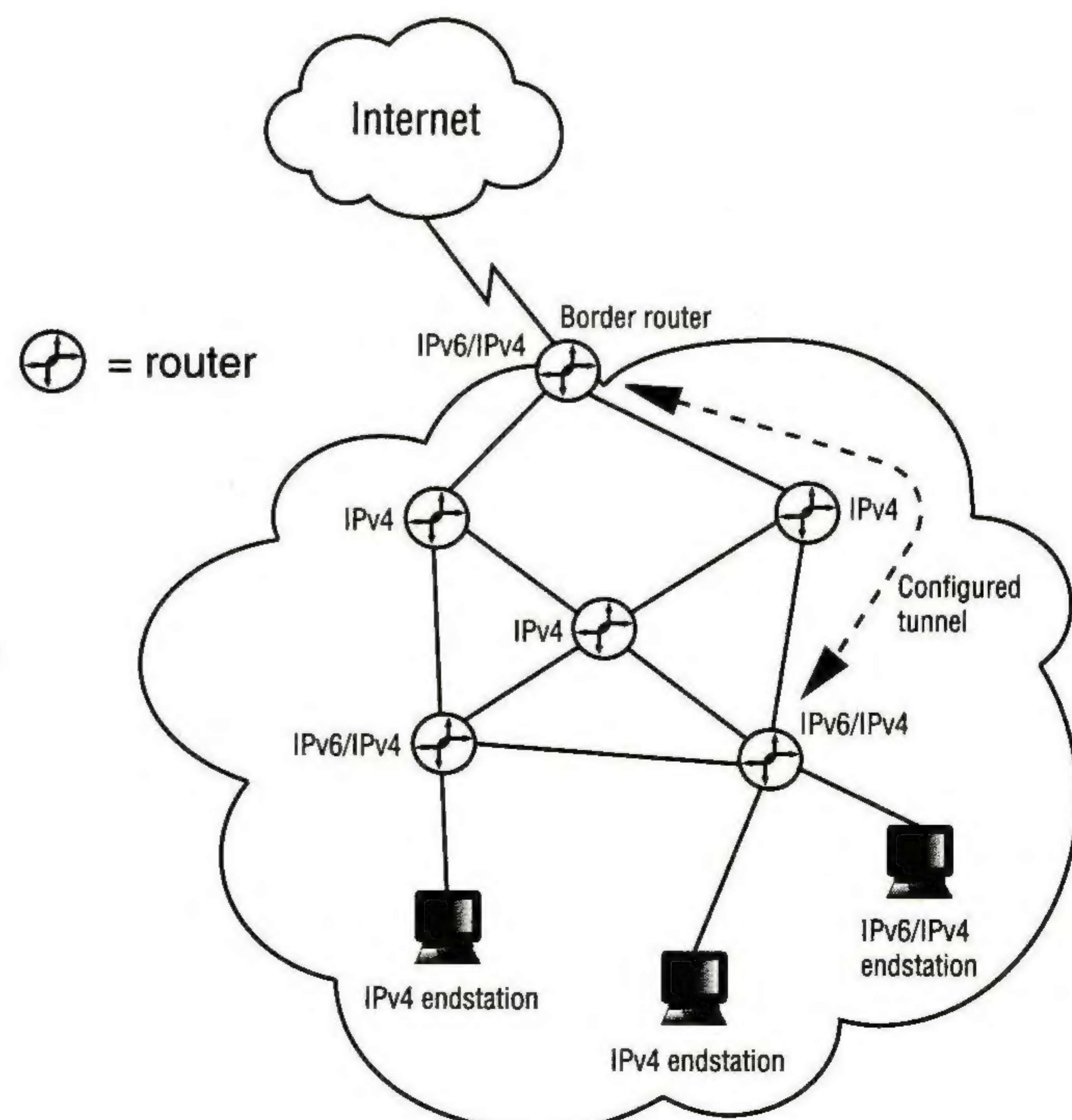


Figure 3: Configured Tunneling

Automatic tunneling uses the same underlying mechanisms as configured tunneling, but eliminates the need to configure each tunnel individually. A special IPv6 address format is defined—the IPv4-compatible address—which holds an IPv4 address in the low-order 32-bits. An IPv4-compatible address is identified by an all-zeros 96-bit prefix, and holds an IPv4 address in the low-order 32 bits. IPv4-compatible addresses are structured as shown in Figure 4.

80 bits	16	32 bits
0000. 0000	0000	IPv4 Address

Figure 4: IPv4-Compatible IPv6 Address Format

An IPv4-compatible address can be viewed as a single address that serves both as IPv6 and IPv4 addresses. The entire 128-bit IPv4-compatible IPv6 address is used as the node's IPv6 address, while the IPv4 address embedded in low-order 32 bits serves as the node's IPv4 address. The embedded IPv4 address is assigned according to the IPv4 addressing plan. Nodes that already have an IPv4 address assignment can use that address in an IPv4-compatible address.

Hosts that engage in automatic tunneling are assigned IPv6 addresses of this form. When an IPv6 node wishes to deliver an IPv6 packet that is addressed to an IPv4-compatible IPv6 address, it can tunnel that packet through the IPv4 routing fabric to the end destination by using the IPv4 address embedded in the IPv6 destination address. Since the tunnel endpoint address is implicit in the compatible destination address of the packet, this form of tunneling is only used when sending packets to their final end destination. So automatic tunneling is typically used to send packets to hosts, not for links between routers (see Figure 5).

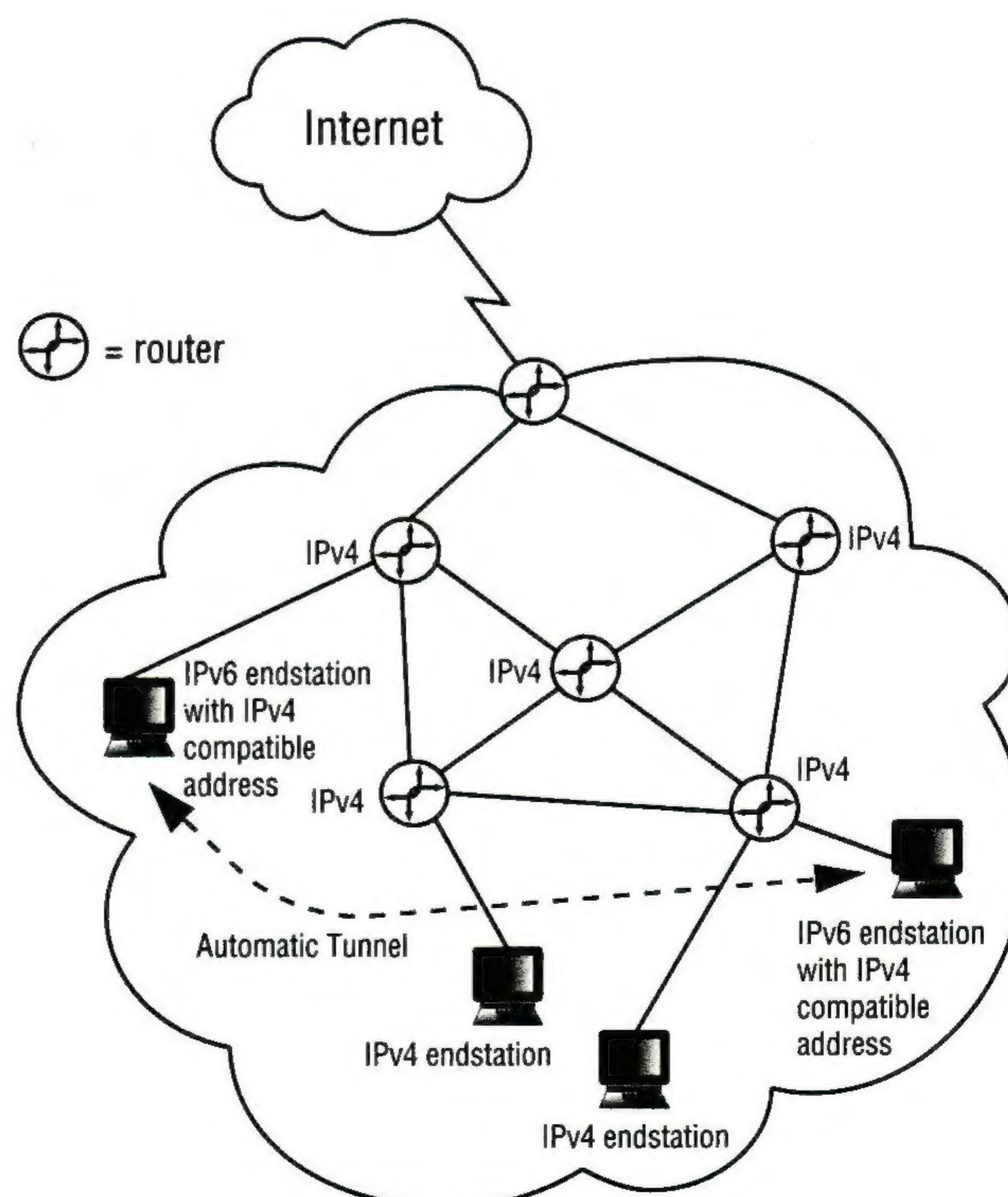


Figure 5: Automatic Tunneling

continued on next page

IPv6 Transition Mechanisms (*continued*)

IPv6/IPv4 nodes that perform automatic tunneling may use IPv4 address configuration protocols such as DHCP, BOOTP or RARP to learn their IPv4-compatible IPv6 addresses. They do this by simply prepending the 96-bit all-zeros prefix 0:0:0:0:0:0: to the IPv4 address that they acquire via the IPv4 configuration protocol. This mode of configuration allows IPv6/IPv4 nodes to “leverage” the installed base of IPv4 address configuration servers. It can be particularly useful in environments where IPv6 routers and address configuration servers have not yet been deployed.

Applied tunneling

Because both hosts and routers can play the tunnel endpoint role, there are a number of feasible configurations for tunneling. Some of these configurations lend themselves to automatic tunneling while others require the configured tunneling method.

- *Router-to-Router*: IPv6/IPv4 routers interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. In this case, the tunnel spans one segment of the end-to-end path that the IPv6 packet takes.
- *Host-to-Router*: IPv6/IPv4 hosts can tunnel IPv6 packets to an intermediary IPv6/IPv4 router that is reachable via an IPv4 infrastructure. This type of tunnel spans the first segment of the packet’s end-to-end path.
- *Host-to-Host*: IPv6/IPv4 hosts that are interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. In this case, the tunnel spans the entire end-to-end path that the packet takes.
- *Router-to-Host*: IPv6/IPv4 routers can tunnel IPv6 packets to their final destination IPv6/IPv4 host. This tunnel spans only the last segment of the end-to-end path.

In the first two tunneling methods listed above—router-to-router and host-to-router—the IPv6 packet is being tunneled to a router. These configurations typically use configured tunneling. The tunnel endpoint is most likely an intermediary router which must decapsulate the IPv6 packet and forward it on to its final destination. When tunneling to a router, the endpoint of the tunnel is different from the ultimate destination so the addresses in the IPv6 packets being tunneled do not provide the IPv4 address of the tunnel endpoint. Instead, the tunnel endpoint address must be determined from configuration information on the node performing the tunneling.

For each configured tunnel, the encapsulating node must store the tunnel endpoint address. When an IPv6 packet is transmitted over a tunnel, the tunnel endpoint address is used as the destination address for the encapsulating IPv4 header. The determination of which packets to tunnel is usually made by routing information on the encapsulating node. This is usually done via a routing table, which directs packets based on their destination address using the prefix mask and match technique.

Default configured tunnel

IPv6 or dual nodes that are connected to IPv4 routing infrastructures may use a configured tunnel to reach an IPv6 “backbone.” If the IPv4 address of an IPv6/IPv4 router bordering the backbone is known, a tunnel can be configured to that router. This tunnel can be configured into the routing table as a “default route.”

That is, all IPv6 destination addresses will match the route and could potentially traverse the tunnel. Since the “mask length” of such default route is zero, it will be used only if there are no other routes with a longer mask that match the destination.

The tunnel endpoint address of such a default tunnel could be the IPv4 address of one IPv6/IPv4 router at the border of the IPv6 backbone. Alternatively, the tunnel endpoint could be an IPv4 “anycast address.” With this approach, multiple IPv6/IPv4 routers at the border advertise IPv4 reachability to the same IPv4 address. All of these routers accept packets to this address as their own, and will decapsulate IPv6 packets tunneled to this address. When an IPv6/IPv4 node sends an encapsulated packet to this address, it will be delivered to only one of the border routers, but the sending node will not know which one. The IPv4 routing system will generally carry the traffic to the closest router.

Using a default tunnel to an IPv4 “anycast address” provides a high degree of robustness since multiple border routers can be provided, and, using the normal fallback mechanisms of IPv4 routing, traffic will automatically switch to another router when one goes down.

Automatic tunneling

When a host is the target endpoint (in host-to-host and router-to-host tunneling), IPv6 packets are tunneled all the way to the final destination. These configurations typically use automatic tunneling. In this case, the tunnel endpoint is the node to which the IPv6 packet is addressed. Since the endpoint of the tunnel is the ultimate destination of the IPv6 packet, the tunnel endpoint can be determined from the destination IPv6 address of that packet: If that address is IPv4-compatible, then the low-order 32 bits hold the IPv4 address of the destination node, and that can be used as the tunnel endpoint address. Hence automatic tunneling avoids the need to explicitly configure the tunnel endpoint address. IPv6 packets that are not addressed to an IPv4-compatible address cannot use automatic tunneling.

IPv6/IPv4 nodes need to determine which IPv6 packets can be sent via automatic tunneling. One technique is to use the IPv6 routing table to direct automatic tunneling. An implementation can have a special static routing table entry for the prefix 0:0:0:0:0:0/96 (that is, a route to the all-zeros prefix with a 96-bit mask). Packets that match this prefix are sent to a pseudo-interface driver which performs automatic tunneling. Since all IPv4-compatible IPv6 addresses will match this prefix, all packets to those destinations will be auto-tunneled (unless a better match route is available).

Tunneling and DNS

When an IPv4-compatible IPv6 address is assigned to an IPv6/IPv4 host that supports automatic tunneling, the corresponding A and AAAA records can be listed in the DNS. The AAAA record holds the full IPv4-compatible IPv6 address, while the A record holds the low-order 32 bits of that address. The AAAA record will be located by queries from IPv6 hosts, while the A record will be found by queries from IPv4-only hosts.

The decision to store an AAAA record holding an IPv4-compatible address in the DNS for an IPv6 host can be used as a policy control for traffic to that host. If an IPv4-compatible address is listed, then other hosts will originate tunneled traffic to that host. If only an A record is listed, then the host will “appear” to others to be an IPv4-only host, and only IPv4 traffic will be sent.

IPv6 Transition Mechanisms (*continued*)

When a query from an IPv6/IPv4 host locates an AAAA record holding an IPv4-compatible IPv6 address, as well as an A record holding the corresponding IPv4 address, the resolver library need not necessarily return both addresses to the application. It has three options:

- Return only the IPv6 address to the application
- Return only the IPv4 address to the application
- Return both addresses to the application

The determination of which address to return can be used as a policy switch on the host to control the type of traffic originated from that host. If the system administrator wishes to prevent that host from originating tunneled traffic, he or she can configure the resolver library to return only IPv4 addresses to the application. If tunneled traffic is permissible, then the administrator can allow IPv6 addresses (and hence IPv4-compatible addresses) to be returned to applications.

Tunneling implementation issues

Nodes that perform tunneling need to deal with a few implementation issues, which are common to both automatic and configured tunneling. Many of these issues relate to the fact that the topology and routing operations of an IPv4 tunnel is largely transparent to the IPv6 nodes using it.

- *Tunnel MTU and fragmentation*: It is technically feasible for an encapsulating node to treat a tunnel as a virtual IPv6 link with a large *Maximum Transmission Unit* (MTU), relying on IPv4 layer fragmentation and re-assembly to deliver IPv6 packets that are larger than the MTU of the underlying links of the path between the encapsulating and decapsulating node. But this would result in fragmentation inside the tunnel, which is inefficient. A better approach is for the encapsulating node to perform IPv4 path MTU discovery of the tunnel path, and then use the IPv6 path MTU discovery to report the MTU of the tunnel back to the originating host.

- *Maintaining tunnel state information*: If the encapsulating node performs IPv4 path MTU discovery on its tunnels, it will need to maintain state information for each tunnel. Since the number of tunnels that a node may be using may grow to be quite large, this node should employ a scheme to cache the state information it needs, and periodically discard state information that is not being used.

- *IPv6 hop limit and IPv4 TTL*: IPv6 tunnels over IPv4 are treated as “single hop” links from the IPv6 perspective. That is, the IPv6 hop limit is decremented by one when an IPv6 packet traverses a tunnel. But the encapsulating node must use a *Time To Live* (TTL) value in the encapsulating IPv4 header that is large enough to guarantee that the encapsulated packet will not expire in the tunnel, enroute to the decapsulating node.

- *IPv4 ICMP error messages and tunneling*: Tunneled packets may fail to be delivered to the tunnel endpoint because the tunnel endpoint is unreachable, the IPv4 TTL is not large enough, or the packet is too big. These failures will elicit IPv4 *Internet Control Message Protocol* (ICMP) error messages, directed back to the tunnel entry point, from IPv4 routers along the tunnel path. Some of these ICMP error messages may not contain enough of the original IPv6 packet to identify its source, since many IPv4 routers return only 8 bytes of data beyond the IPv4 header of the packet in error. Other IPv4 routers return much more data. If possible, the encapsulating node should attempt to recover the original IPv6 packet, and generate the appropriate IPv6 ICMP error message back to the originating node.

**Tunneling and routing
in depth**

During an extended IPv4-to-IPv6 transition period, both IPv4 and IPv6 routing infrastructure will be present. Earlier we discussed how in the basic pure dual-IP layer operation, routing within the IPv4 infrastructure may be essentially independent of routing within the IPv6 infrastructure. However, at least initially, IPv6-capable domains might not be globally interconnected via IPv6-capable Internet infrastructure and therefore may need to communicate by using tunneling across IPv4-only backbone networks.

In order to achieve dynamic routing in such a mixed environment, there need to be mechanisms to globally distribute IPv6 network-layer reachability information to routing nodes in dispersed IPv6 domains. (Alternatively, some of the same techniques might be used in later stages of the transition to route IPv4 packets between isolated IPv4-only networks over IPv6 backbones.)

Use of tunneling requires consistency of routes between IPv4 and IPv6. For example, consider a packet which starts off as an IPv6 packet, but then passes through an IPv4 tunnel (i.e., is encapsulated in an IPv4 packet) in the middle of its path from source to destination. This packet must be routed (using IPv6 routing) to the correct initial tunnel endpoint, traverse the tunnel as an IPv4 packet, and then traverse the remainder of the path again as an IPv6 packet. Clearly this packet has to follow a consistent route for the entire path from source to destination. The implications of this process on routing are discussed separately for router-to-router tunnels, host-to-host tunnels, host-to-router tunnels, and router-to-host tunnels.

**Router-to-router
tunnels**

Router-to-router tunnels are based on manual configuration of both ends of the tunnel. Specifically, the router at each end of the tunnel must be manually configured to know the addresses associated with the other end of the link. Such tunnels are also referred to as “fully manually configured” tunnels, since both ends of the link must be configured.

In router-to-router tunnels handling IPv6 over IPv4, routers treat the link as they would any other normal point-to-point link. For example, dynamic routing protocols such as OSPF or BGP/IDRP may send reachability information over this link as they would over any other type of link. The decision to forward a packet over a manually-configured router-to-router tunnel is therefore made in the same manner as the decision to forward a packet over any other type of link. Specifically, packets are forwarded based on the routes computed by standard routing protocols. These routes may use normal links and tunneled links in any combination.

Use of router-to-router manually configured tunnels has the advantage that the underlying infrastructure is transparent to the protocols that are forwarded over the tunnel. For example, if IPv6 is tunneled over IPv4, then the IPv4 infrastructure is used for forwarding IPv6, but the internal details of the IPv4 infrastructure is of no concern to IPv6 routers and IPv6 routing protocols. Also, all types of IPv6 addresses without exception can be advertised in IPv6 routing and tunneled over IPv4 networks. Since IPv6 packets are encapsulated only when they travel over network segments that do not support IPv6, and are forwarded according to their native headers elsewhere, this method does not constrain the types of policy routing which may be employed over the IPv6 portion of the data path.

IPv6 Transition Mechanisms (*continued*)

However, a router-to-router manually configured tunnel differs from a normal link in one important aspect: in many cases it is likely to have lower performance, such as lower throughput or greater delay. The use of a routing protocol such as RIP, which treats each link as being equal, could lead to sub-optimal routes. However, this is not a problem with more flexible routing protocols such as OSPF, which allows a wide dynamic range in the metrics assigned to each link. Specifically with OSPF, the tunneled link could be given a higher cost when compared to other links.

Beyond router-to-router configured tunnels, tunnels in which both ends of the link are manually configured can, in principle, also be used from host to router or from host to host. However, when a number of hosts are involved as endpoints, the requirement that each end of the tunnel be configured makes “fully manual” tunnels less useful than it is for routers.

Host-to-host automatic tunneling

If both source and destination hosts make use of IPv4-compatible IPv6 addresses, then it is possible for automatic tunneling to be used for the entire path from the source host to the destination host. In this case, the IPv6 packet is encapsulated in an IPv4 packet by the source host, and is forwarded by routers as an IPv4 packet all the way to the destination host. As discussed earlier, this feature allows initial deployment of IPv6-capable hosts before any routers are updated.

A source host may make use of host-to-host automatic tunneling provided that *all* of the following are true:

- The source address is an IPv4-compatible IPv6 address
- The destination address is an IPv4-compatible IPv6 address
- The source host doesn't know of any neighboring IPv6-capable router
- The source host does know of one or more neighboring IPv4-capable routers

If all of these requirements are true, then the source host may encapsulate the IPv6 packet in an IPv4 packet, using a source IPv4 address extracted from the associated source IPv6 address, and a destination IPv4 address extracted from the associated destination IPv6 address. Where host-to-host automatic tunneling is used, the packet is forwarded as a normal IPv4 packet for its entire path, and is decapsulated (i.e., the IPv4 header is removed) only by the destination host. Host-to-host automatic tunneling requires that normal IPv4 routing be operational, but makes no requirements whatsoever on IPv6 routing.

Host-to-router tunnels

In some cases a dual-IP layer host may need to transmit an IPv6 packet, but may have no local IPv6-capable router that it can use for this purpose. Instead, the host may use tunneling to an IPv6-capable router. This capability allows the host to transmit packets through an arbitrary number of IPv4 nodes to an IPv6-capable backbone, which will then in turn transmit the packets using normal IPv6 forwarding. Host-to-router tunnels may be accomplished by manually configuring the dual-IP layer host with an IPv4 address that it can use to reach the IPv6 backbone.

For the conversation to work in both directions it is necessary for a router-to-host tunnel to have a return path. This requires either that both ends of the tunnel be manually configured, or that automatic tunneling be used in the backward (router-to-host) direction. This latter type of tunnel may be referred to as a “half manually configured” tunnel (see Figure 6), since manual configuration is used in one direction, but automatic tunneling is used in the other.

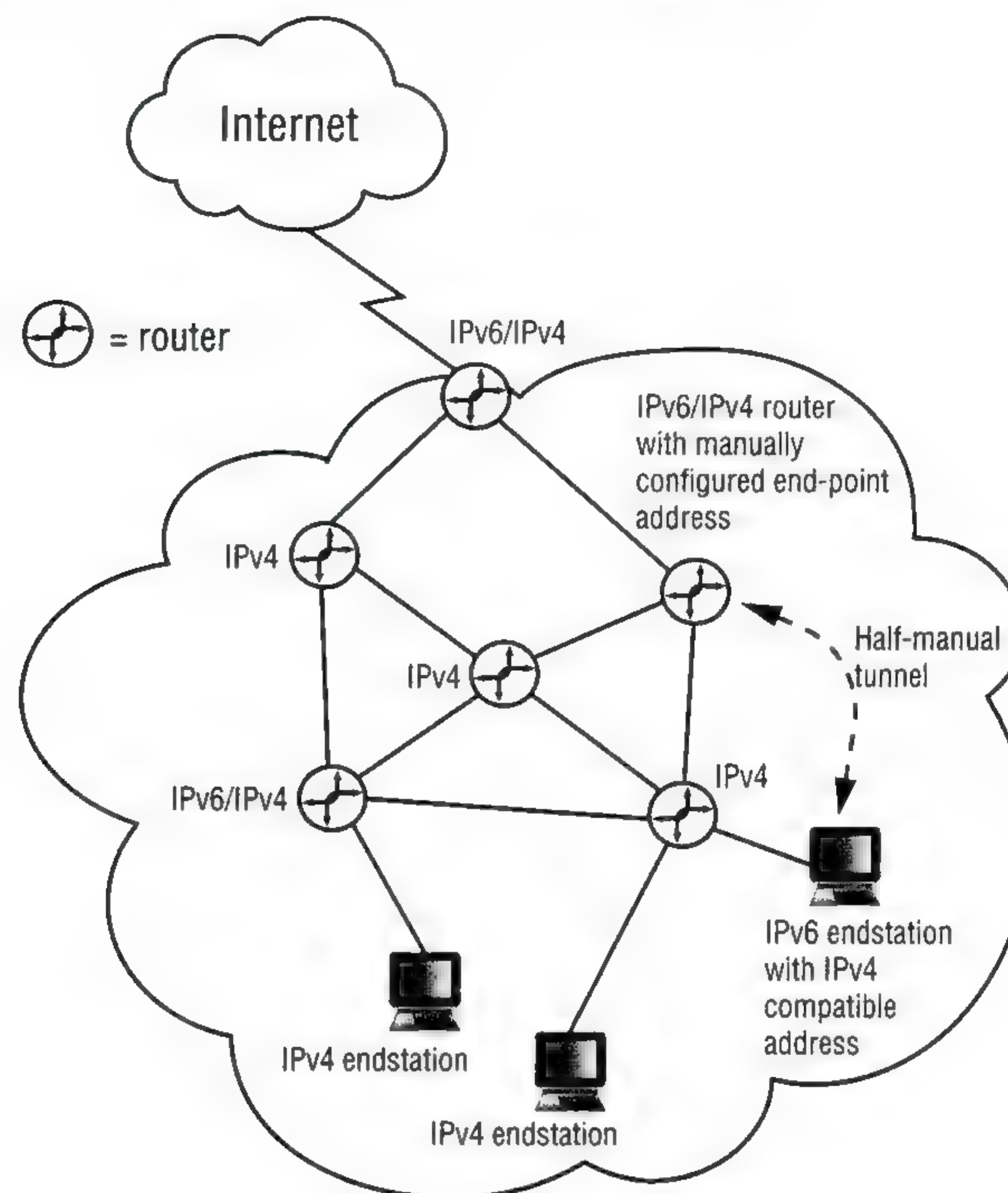


Figure 6: Half-manual Tunnel

A half-manually configured tunnel occurs when the host is configured to know how to find the router, but the router is not configured with any specific knowledge of the host and will use automatic tunneling to find it. This, of course, requires that the host have an IPv4-compatible IPv6 address and that the host be configured with an IPv4 address to use for tunneling to the IPv6-capable router.

A source host may make use of host-to-router half-manually configured tunneling provided that *all* of the following are true:

- The source address is an IPv4-compatible IPv6 address.
- The source host does not know of any neighboring IPv6-capable router.
- The source host does know of one or more neighboring IPv4-capable routers.
- The source host is configured with an IPv4 address of a router which can serve as the tunnel endpoint.
- The destination address is *not* IPv4-compatible (if the destination address is IPv4-compatible, then host-to-host automatic tunneling may be used instead).

If all of these requirements are true, then the source host may encapsulate the IPv6 packet in an IPv4 packet, using a source IPv4 address that is extracted from the associated source IPv6 address, and a destination IP address that corresponds to the configured address of the dual router which is serving as the tunnel endpoint.

continued on next page

IPv6 Transition Mechanisms (*continued*)

Reachability for host-to-router tunnels

The dual router which is serving as the end point of the half-manual tunnel must advertise reachability into IPv4 routing sufficient to cause the encapsulated packet to be forwarded to it. The simplest approach is for a single IPv4 address to be assigned to the router for use as a tunnel endpoint. A tunneling dual router with connectivity to the IPv6 backbone can advertise a host route to this address (into the IPv4-only network). Each dual host in the associated IPv4-only network is configured with the address of this tunnel endpoint.

In some cases there may be multiple dual routers which can serve as endpoints for automatic tunneling to hosts from any one IPv4-only network. In this case, again, each host may be configured with a single address representing the tunnel endpoint. However, all dual routers with connectivity to the IPv6 backbone that are capable of serving as endpoints for the automatic tunnels from this region may advertise a host route to the associated IPv4 address. This allows encapsulating packets using host-to-router tunneling to be forwarded to the tunnel endpoint that is selected by the local routing policy (in general this will be the nearest dual router). In this case the one IPv4 address is operating as an “anycast” address, since it allows the tunneled packets to be delivered to any one of the multiple dual routers.

Finally, in some cases there may be some reason for specific hosts to prefer one of several tunnel endpoints, while allowing all potential tunnel endpoints to serve as backups in case the preferred endpoint is not reachable. In this case, each dual router with IPv6 backbone connectivity that is serving as a potential tunnel endpoint is given a unique IPv4 address taken from a single IPv4 address block. (For example, if there are less than 255 such dual routers, a single class C IPv4 network number may be used). Each dual router then advertises two routes into the IPv4 network: a host route corresponding to the tunnel endpoint address specifically assigned to it, and also a network route to the associated IPv4 address block (e.g., to the class C network in the normal case).

Each dual host in the IPv4-only region is configured with an IPv4 address corresponding to the preferred tunnel endpoint. If the associated dual router is operating, then the packet will be delivered to it based upon the associated host route. However, if the associated dual router is down, but some other dual router serving as potential tunnel endpoint is operating, then the packet will be delivered to the nearest operating tunnel endpoint.

Router-to-host automatic tunneling

Clearly if tunneling is used from a host to a backbone IPv6 router, it is also necessary to be able to use tunneling from the router to the host. In this case (provided that the destination host has an IPv4-compatible IPv6 address) normal IPv6 forwarding may be used for part of the packet's path, and router-to-host automatic tunneling may be used to get the packet from an encapsulating dual router to the destination host.

Normal packet forwarding is straightforward in this case: the encapsulating router creates the encapsulating IPv4 header using an IPv4 address assigned to itself as the source IPv4 address, and using a destination IPv4 address extracted from the destination IPv4-compatible IPv6 address. The encapsulated packet is forwarded from the encapsulating router to the destination host using normal IPv4 routing.

In this case, the challenging part is the IPv6 routing required to deliver the IPv6 packet from the source host to the encapsulating router. For this to happen, the encapsulating router has to advertise reachability for the appropriate IPv4-compatible IPv6 addresses into the IPv6 network.

Router-to-host tunneling typically occurs when one or more dual-IP layer routers are sitting on the boundary between an IPv4-only network and a dual-IP layer network. In this case, these "border routers" need to advertise into IPv6 routing (in the dual network) that they can reach certain IPv4-compatible IPv6 addresses corresponding to the addresses that exist in the IPv4 network. In general this requires manual configuration of the border routers. However, in most cases it may require only one or a small number of address prefixes be advertised for the entire local IPv4 network. This is, therefore, likely to represent much less configuration than individually configuring router-to-host links.

References

- [1] Bradner, S., & Mankin, A., "The Recommendation for the IP Next Generation Protocol," RFC 1752, January 1995.
- [2] Postel, J., "Assigned Numbers," RFC 1700, October 1994.
- [3] Hinden, R., Editor, "IP Version 6 Addressing Architecture," Work in Progress, April 1995.
- [4] Rekhter, Y., & Li, T., "An architecture for IPv6 Unicast Address Allocation," Work in Progress, March 1995.
- [5] Rekhter, Y., & Lothberg, P., "An IPv6 Global Unicast Address Format," Work in Progress, March 1995.
- [6] Gilligan, R. E., Thomson, S., & Bound, J., "IPv6 Program Interfaces for BSD Systems," Work in Progress, July 1995.
- [7] Hinden, R. "IP Next Generation Overview," *ConneXions*, Volume 9, No. 3, March 1995.
- [8] *ConneXions*, Volume 8, No. 5, May 1994, Special Issue on IP The Next Generation.
- [9] Fuller, V., Li, T., Yu, J., Varadhan, K., "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," RFC 1519, September 1993.
- [10] Fleischman, E., "A User's View of the Next Generation of IP (IPng)," *ConneXions*, Volume 8, No. 5, May 1994.
- [11] Rekhter, Y., Li, T., "Address Ownership Considered Fatal," *ConneXions*, Volume 9, No. 7, July 1995.

ROSS CALLON is a consulting engineer with Bay Networks Incorporated in Billerica, Massachusetts, and has more than 15 years' experience in data communications. He was the original proposer of a dual stack transition scheme and of TUBA, as well as co-author of the NSAP Guidelines standard in the Internet Engineering Task Force (IETF). He is co-chair of the IETF IPng Working Group, and was a member of the IPng Directorate. He is also a regular contributor to the ATM Forum, and has been involved in the ATM PNNI routing and multi-protocol over ATM efforts. E-mail: rcallon@baynetworks.com

ROBERT E. GILLIGAN is a staff scientist at Sun Microsystems, Inc. where he is responsible for developing the TCP/IP software in Solaris 2. Prior to joining Sun, he worked on Internet and Packet Radio projects at SRI International. Mr. Gilligan is also an active participant in the IETF. E-mail: Bob.Gilligan@eng.sun.com

[Ed.: This article is adapted from the book *IPng: Internet Protocol Next Generation*, Edited by Scott O. Bradner and Allison Mankin, published by Addison-Wesley, ISBN 0-201-63395-7, 1995. Used with permission].

Collaborative Virtual Environments on the Internet

by Steve Benford and Chris Greenhalgh,
The University of Nottingham

Introduction

Recently, several research laboratories have begun experimenting with *Collaborative Virtual Environments* (CVEs); distributed multi-participant virtual reality systems that provide direct support for cooperative working. This experimentation has reached the stage of holding initial virtual meetings across the Internet. One recent example involved nine simultaneous participants from five organisations distributed across three countries and lasted for an hour and a half.

This article provides a brief introduction to Collaborative Virtual Environments, touching on the motivations behind them, the current state of the art and key research issues. As a concrete example, it also describes a specific CVE called MASSIVE which has been developed by the authors and used for recent experimentation. The aim of the article is therefore to provide an overview of what is a rapidly expanding area of interest for networking researchers, providers and users alike.

Motivation

The essence of a Collaborative Virtual Environment is that several participants share and freely navigate a computer generated 3-D graphical space. Furthermore, they are embodied within this space (i.e., directly represented to one another in a graphical form), and are able to communicate over various media. They should also be able to interact with other objects inhabiting the space. There have been three main motivations behind the development of CVEs:

- *Addressing the limitations of current tele-conferencing technologies:* There is considerable evidence to suggest that spatial cues such as body orientation and gaze direction play a critical role in the fluid management of conversational turn-taking. However, current tele-conferencing systems (especially video conferencing) do not support such cues. There is no way to gaze at a particular participant in most multi-party video conferences. Instead, glancing at the camera has the apparent effect of gazing at everyone in the conference. In CVEs however, participants have a common understanding of each others' viewpoints within a shared space and can infer what others are seeing from their embodiments (this is not the same as actually seeing what others are seeing!).
- *Integration with electronic information:* As information continues to migrate into electronic form, so there will be an increasing need to integrate electronic representations of data into meetings. CVEs allow participants to directly share their meeting space with the information they are discussing. In other words, as well as interacting with one another, participants can interact with 3-D information visualisations including models of real-world objects (e.g., buildings, landscapes and engineering designs) and representations of more abstract information (e.g., electronic documents, complex software systems, scientific visualisations, statistical and financial information and even the World-Wide Web).
- *Scalability:* Real world "meetings" often involve many hundreds or even thousands of simultaneous participants. For example, consider the kinds of interactions that occur at large conferences, trade shows and exhibitions. In addition to formal presentations, mingling and browsing situations are an important part of such events and provide an opportunity for many business relationships to be forged.

Applications for professionals and citizens alike

However, current real-time conferencing technologies appear to scale no further than a few participants and cannot support these less formally structured kinds of communication (on the other hand, we are all familiar with asynchronous technologies that do scale to very large numbers of participants). We argue that, the additional screen real-estate provided by 3-D graphical environments coupled with the ability to freely navigate them means that CVEs may be inherently more scalable and may potentially support a wide range of meeting scenarios than do current technologies.

One can imagine many future business applications of CVEs including general meeting support, education and training, simulation and mission rehearsal, shared design, scientific visualisation and concurrent engineering. However, further thought uncovers further potential applications for the general citizen. As well as the obvious case of interactive multi-participant computer games, one can envisage a wide range of future mass participation electronic events spanning both arts and sports, possibly involving hundreds of thousands of simultaneous participants. The role of the crowd in adding excitement to many real-world events is well known. There is a great difference between watching an event on TV and experiencing it from within the crowd. One might also argue that, just as the simplicity of the World-Wide Web interface has opened up the information publishing and retrieval aspects of the Internet to the general citizen, so the natural spatial metaphor of CVEs might open up the real time communication aspects.

Predicting the future is, of course, a difficult game. However, one can already see how the necessary infrastructure to support future citizen applications of CVEs might emerge. There is a growth and merging of interests in applications such as interactive and digital TV, games and tele-shopping, fuelled by the spread of networking to the home and by the development of low cost domestic computing devices (advanced games consoles or "set top boxes"). Such emerging infrastructure may turn out to be an ideal platform for deploying large scale CVEs.

State of the art

Many researchers world-wide have been experimenting with Collaborative Virtual Environments. Within Europe, and especially within the UK, several consortia have been funded to develop CVEs to support business and professional users. For example, the European COMIC project has recently brought together researchers from Computer Science, Psychology and Sociology to conduct basic research into CVEs [1]. The UK government is currently funding the *Virtuosi* project, a three year pilot involving universities, telecommunications companies, VR manufacturers and end users which aims to construct and evaluate two pilot CVE applications, the *Virtual Factory* for the engineering industry and the *Virtual Catwalk* for the fashion industry. The European Commission has also recently announced the COVEN project, a three year European wide pilot funded under the Advanced Communication Technologies (ACTS) programme. Probably the most influential single project in this area has been *DIVE*, a development of the Swedish Institute of Computer Science (SICS) which has resulted in a freely available Internet based CVE which has been used world-wide as the basis for a host of other projects [2]. [DIVE provides a general development environment for Internet based CVEs and more details can be obtained from Lennart Fahlén (lef@sics.se)].

Collaborative Virtual Environments (*continued*)

US effort has primarily focused on military applications. Specifically, the US Navy has developed a large scale military simulator capable of supporting of the order of a hundred simultaneous users called *NPS-NET* [4]. A notable feature of NPSNET has been the development of networking techniques which involve the scoping of participants interaction according to a cellular division of space and the mapping of this onto underlying multicast protocols in order to minimise network traffic. NPSNET has also been based on a emerging ARPA military standard called the *Distributed Interactive Simulation* (DIS) standard, which is becoming an increasing focus of attention (see below).

Japanese activity has considered both business and citizen applications of CVEs. The former includes the *Collaborative Workspace* from NTT [6], an immersive conferencing system supporting real-time facial action capture and animation (as opposed to the use of video). An example of the latter has been NTT's *Interspace* project, a demonstrator which supports groups of people navigating a virtual city in order to go shopping and engage in recreational activities [5]. Finally, the *Greenspace* project is worthy of note as it represents a recent inter-continental demonstration of a simple CVE between research laboratories in the US and Japan.

These various research projects represent first steps in exploring the potential of CVEs and in developing the necessary supporting techniques. It should also be noted that most of the current generation of commercial VR products provide some limited support for networking. Examples of such products include *dVS* from Division Ltd., *Elysium* from Virtuality, *Superscape* from Superscape Ltd., and the *World Toolkit* from The Sense 8 Corporation, all of which offer a limited networking capability.

Research issues

So far, we have presented the background to CVEs in terms of motivations and a quick summary of the state of the art. Next, we turn our attention to some of the technical issues raised by CVEs that must be addressed in order for their widespread application to become possible. From a user's point of view, a CVE involves a number of participants entering a shared virtual space where they can communicate with one another and interact with various other objects. This raises several interesting issues:

- *How are users represented to one another?* User embodiments might be designed to convey many features of an individual's identity and activity including: who they are, where they are located, where they are looking, what they are manipulating, their natural gestures and facial expressions, where they have been, the extent to which they are actually present and so on. There are also many possible techniques available to support these. For example, just considering the issue of conveying facial expression, one might texture map live video onto a virtual body or one might build an expression recognition system and then reproduce a graphical equivalent expression as in [6]. The trade-offs between these different issues and techniques may be complex and also application specific, and considerable research is needed to understand them in detail.
- *How is the space defined?* The nature of a space undoubtedly affects the kinds of communication that happens within it. To what extent can we borrow from existing disciplines of architecture and planning when designing virtual meeting spaces? Alternatively, should we follow the route of building more abstract spaces based on data visualisation techniques? If so, what techniques are available to us?

- *What additional communication mechanisms are required?* Users require mechanisms for managing their interaction with one another, particularly in large scale meetings where it is not possible to perceive everything that everyone says. In contrast to traditional floor control mechanisms, what spatial techniques might be appropriate? In particular, we need to consider how users can manage their connectivity to each other as they move about (dynamic meeting configuration) and how they can easily manage the quality of service across different connections (e.g., so that nearby people are perceived in detail and more distant people only fuzzily).

Underlying these issues is a further set of questions concerning network support for CVEs. For example, how can we map different regions of a densely populated virtual world onto underlying network multicast groups or how can we prioritise the traffic generated by different events in a virtual world so as to maintain a user's sense of "presence" when the network becomes congested?

Example: the MASSIVE system

In this section, we describe one specific example of a Collaborative Virtual Environment, the MASSIVE system [3], in order to show how some of the above research issues have been addressed in our own work.

MASSIVE has been built at the University of Nottingham with a view to supporting virtual tele-conferencing across the Internet. Within any given instantiation of the system, the MASSIVE universe is structured as a set of virtual worlds connected via *portals*. Each world defines a disjoint and infinitely large virtual space which may be inhabited by many concurrent users. Portals allow users to jump from one world to another.

Users can interact with one another over combinations of graphics, audio and text media. The graphics interface renders objects visible in a 3-D space and allows users to navigate this space with a full six degrees of freedom. The audio interface allows users to hear objects and supports both real-time conversation and playback of pre-programmed sounds. The text interface provides a MUD-like view of the world via a window (or map) which looks down onto an infinite 2-D plane across which users move (similar in style to the UNIX games *Rogue* and *Nethack*). Text users are embodied using a few text characters and may interact by typing text messages to one another or by "emoting" (e.g., smile, grimace etc.).

A key feature of MASSIVE is that these three kinds of interface may be arbitrarily combined according to the capabilities of a users terminal equipment. Thus, at one extreme, the user of a sophisticated graphics workstation may simultaneously run the graphics, audio and text clients, the latter being slaved to the graphics client in order to provide a map facility and to allow interaction with non-audio users. At the other extreme, the user of a dumb terminal (e.g., a VT-100) may run the text client alone. It is also possible to combine the text and audio clients without the graphics client and so on.

In order to allow interaction between these different clients a text user may export a graphics body into the graphics medium even though they cannot see it themselves. Similarly, a graphics user may export a text body into the text medium. In other words, text users can be embodied in the graphics medium and graphics users can be embodied in the text medium.

Collaborative Virtual Environments (*continued*)

Aura, awareness, focus, nimbus and adapters

MASSIVE uses a dynamic brokering mechanism to determine whether objects have any media in common whenever they meet in space (i.e., on aura collision). The net effect is that users of radically different equipment may interact, albeit in a limited way, within a common virtual world; for example, text users may appear as slow-speaking, slow moving flatlanders to graphics users.

Communication in MASSIVE is controlled through a novel *spatial model of interaction* which allows users to scope their interaction in terms of maintaining only a limited connectivity to others as they move about and also to flexibly control the quality of service parameters across these different connections.

Details of this model are beyond the scope of this article and can be found in [1]. In essence, connectivity is enabled through a concept called *aura*—a personal subspace which scopes one's presence in space. Quality of service is controlled through a concept called *awareness*, a measure of how much each object is aware of each other connected object. In turn, awareness is controlled by two further concepts called *focus* and *nimbus*. A focus represents the spatial allocation of a user's attention and a nimbus the spatial projection of their information. Thus, A's awareness of B is some function of A's focus on B and B's nimbus on A. Finally, various *adapter objects* change a user's aura, focus and nimbus allowing them to reconfigure their communication capabilities (e.g., standing at the virtual podium expands one's aura and nimbus allowing one to address a greater number of people than usual).

The net effect of aura, awareness, focus and nimbus is that users move around a virtual space and use various objects in order to manage connectivity and quality of service. More specifically, audio, textual and graphical representations of other people are sensitive to relative positions and orientations (e.g., people may become quieter as one moves or turns away from them).

Embodiment in MASSIVE

A person's MASSIVE embodiment determines how they appear to other users. Each user may specify their own graphics embodiment in a personal configuration file using a simple geometry description format. In addition, we provide some default graphics embodiments intended to convey the communication capabilities of the users they represent (which is an important issue in a heterogeneous environment). For example, an audio user has ears, a non-immersive (and hence monoscopic) user has a single eye and a text user has the letter "T" embossed on their head. The aim of such embodiments is to provide other users with the necessary basic communication cues to decide how to address them. The basic shape of graphics embodiments is also intended to convey orientation in a simple and efficient manner.

Figure 1 shows a screenshot from MASSIVE of a meeting in progress involving five participants who are using a conference table adapter. In this case, we ("Chris") have adopted a slightly "out of body" virtual camera view so that we are looking over our own shoulder down onto the meeting—MASSIVE offers a range of such views.

Early experiences

MASSIVE has been successfully used to hold a number of virtual meetings across the Internet. The most successful of these took place on March 28th, 1995 and involved nine participants distributed across five organisations from three countries (the UK, Sweden and Germany).

All of the participants were audio/graphical and the whole meeting lasted for an hour and a half. As a result of this and other meetings, we have gathered some initial feeling for both the advantages of this technology, and also some of the issues that must be addressed in its future development.

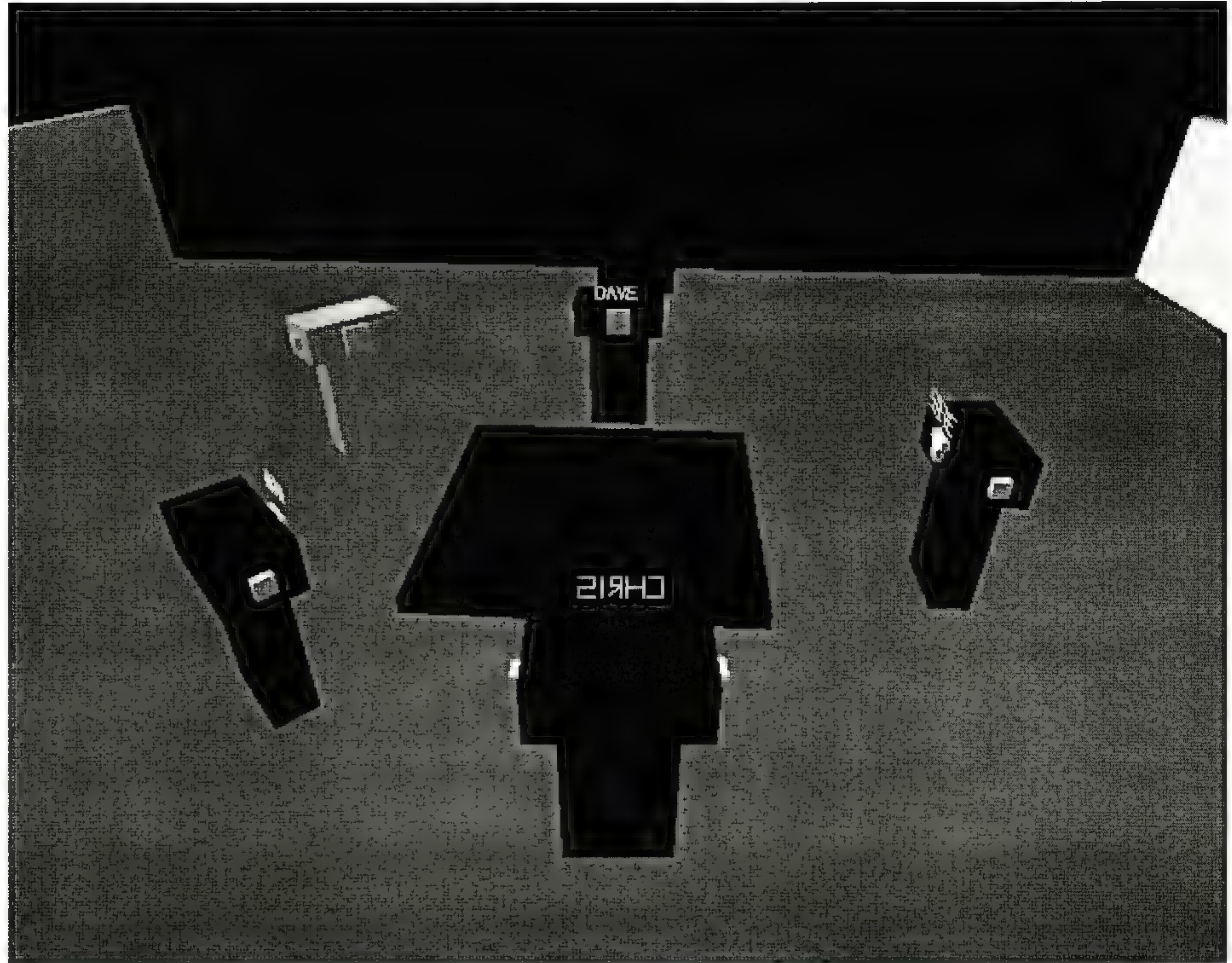


Figure 1: MASSIVE users get together

[Ed.: You can see more of these images in glorious color by pointing your browser at <http://www.crg.cs.nott.ac.uk/~cmg/massive.html>.]

The primary advantages are that, on the whole, the technology works. MASSIVE can be installed on standard workstations and seems to support meetings involving roughly ten participants at a time. These meetings have undoubtedly been enjoyable and initial evaluations point to some positive features. First, there is some evidence from conversational analysis of video footage that people coordinate movements of their virtual bodies with their conversation in a similar manner to the real world (at least sometimes). It also seems that novice users learn the system fairly easily (about three hours seems typical).

On the negative side, the current field of view of computer monitors seems to be too limited to provide a powerful sense of peripheral awareness of other people at the edge of one's field of vision. Similarly, the clumsy nature of navigating in 3-D with a 2-D mouse limits one's ability to move quickly and seems to constrain people's ability to perform spatial coordination. The lack of eye-tracking ability also drastically limits any support for gaze direction. Finally, there is an interesting issue concerning the degree of presence of different participants. It seems that, on occasion, participants become distracted by events in the real-world but remain logged in. Their embodiments continue to suggest full presence and several embarrassing incidents have occurred where people have tried to interact with uninhabited embodiments!

Collaborative Virtual Environments (*continued*)

The future

If MASSIVE is, in some ways, representative of the current state of the art of Collaborative Virtual Environments, then what of the future?

First, there are many basic research issues to be addressed before truly useful CVEs can be developed. None of the above mentioned issues has yet been explored to any great depth. Second, there is a clear need to develop applications of CVEs which show benefits to the widest possible range of users. Projects are already underway to demonstrate various industrial applications (e.g., *Virtuosi*). We would argue that equivalent effort should be focused at citizen oriented applications. Third, there is the question of standards. Although one might feel that CVE development is largely at the stage of basic research, there are in fact already two emerging standards of considerable relevance to CVEs.

The *Virtual Reality Modelling Language* (VRML) is being widely discussed as an emerging de facto standard for integrating VR into the World-Wide Web [7]. In its current form, VRML specifies extremely limited functionality in terms of users navigating simple static 3-D scenes. However, there is considerable interest in VRML and there is currently wide ranging discussion about its future development, including the addition of greater interactivity and also multi-participant interaction.

The *Distributed Interaction Simulation standard* (DIS) has been developed by the US Department of Defense in tandem with US industry with a view to standardising support for distributed VR military simulations [8]. At the heart of DIS lies the specification of various *Protocol Data Units* (PDUs) which convey simulation and state information between different hosts on a computer network. Current PDU definitions enable the transfer of basic state information describing various “vehicles,” including information to be used by dead-reckoning algorithms which can make local predictions about the future behaviour of objects that are owned by remote hosts. DIS has already been used as the basis for a variety of simulators, most notably NPSNET [4].

Although both VRML and DIS appear relevant to the future development of general CVEs, it is also clear that they have, of course, been designed with the respective constraints of WWW and real-world battle simulations in mind. It is certainly not clear that they provide an ideal base for supporting more general CVEs. Again, input is clearly needed from other applications.

Summary

In summary then, our aim in this article has been to introduce the subject of Collaborative Virtual Environments, an emerging topic of interest from the research domain. We hope that we have at least succeeded in stimulating interest in this topic and have managed to provide a sufficient technical justification for the basic principles involved. Further, we hope that, through a discussion of our own MASSIVE system, we have managed to convey a sense of the current state of the art and also point towards the future potential of CVE technologies as well as some key issues that need to be addressed.

References

- [1] Steve Benford, John Bowers, Lennart Fahlén, John Mariani and Tom Rodden, "Supporting Co-operative Work in Virtual Environments," *The Computer Journal*, Volume 37, Number 8, Oxford University Press, 1994.
- [2] Fahlén, L. E., Brown C. G., Stahl, O., Carlsson, C., "A Space Based Model for User Interaction in Shared Synthetic Environments," in Proceedings of InterCHI'93, Amsterdam, 1993, ACM Press.
- [3] Greenhalgh, C. and Benford, S., "MASSIVE: A Virtual Reality System for Tele-conferencing," *ACM Transactions on Computer Human Interaction* (TOCHI), ACM Press (in press).
- [4] Macedonia, M. R., Zyda, M. J., Pratt, D. R., Barham, P. T. and Zeswitz, S., "NPSNET: a network software architecture for large scale virtual environments," *Presence*, 3(4), MIT Press, 1994.
- [5] Suzuki, G., "Interspace: Toward Networked Virtual Reality of Cyberspace," Proceedings of Imagina'95, Monte-Carlo, February, 1995, INA.
- [6] Takemura, H., and Kishino, F., "Cooperative Work Environment Using Virtual Workspace," In Proceedings of CSCW'92, Toronto, November 1992, ACM Press.
- [7] Bell, G., Parisi, A., Pesce, M., "The Virtual Reality Modelling Language," Version 1.0 Specification, URL: <http://vrml.wired.com/vrml.tech>
- [8] "Standard for Information Technology, Protocols for Distributed Interactive Simulation," Institute of Electrical and Electronics Engineers, International Standard, ANSI/IEEE Std. 1278-1993, 1993.
- [9] Crowcroft, Jon and Handley, Mark "The World-Wide Web: How Servers Work," *ConneXions*, Volume 9, No. 2, February 1995.
- [10] Berners-Lee, T., R. Cailliau, A. Loutonen, H. F. Nielsen and A. Secret, "The World-Wide Web," *Communications of the ACM*, Volume 37, No. 8, August 1994.

STEVE BENFORD is a Senior Lecturer in Computer Science at the University of Nottingham. Since completing his PhD on the design of distributed directory services in 1989, he has researched a number of topics including group communications, CSCW, Open Distributed Processing and, currently, distributed virtual reality. In his spare time he plays traditional music on guitar and fiddle. E-mail: sdb@cs.nott.ac.uk

CHRIS GREENHALGH is currently studying for his PhD at the University of Nottingham on the topic of large-scale virtual reality conferencing funded by the UK's Engineering and Physical Research Council (EPSRC). His broader interests include computer mediated communications, CSCW, human computer interaction and natural language processing. Before this, he worked as a Research Engineer at the GEC Hirst Research Centre (UK). E-mail: cmg@cs.nott.ac.uk

Authors' address

Department of Computer Science
 The University of Nottingham
 University Park
 Nottingham NG7 2RD
 United Kingdom
 Tel: +44 115 951 4203 • Fax: +44 115 951 4254
<http://www.crg.cs.nott.ac.uk/>

Call for Papers

A new journal published in cooperation with the ACM called *Wireless Networks* will produce a Special Issue on Mobility and Security in the last quarter of 1996.

Scope Mobility introduces a new dimension to the problem of secure computing and communication. The securing becomes harder and often more important. This is sometimes due to the mobility of the communication devices, sometimes due to the mobility of users (without mobile device), or the mobility of objects, or that of the attackers.

Topics Papers are sought that address the requirements, designs, algorithms and implementation experience for securing networks, distributed systems, information, and applications in environments that can support mobility. Possible topics include, but are not limited to:

- Securing communication and distributed systems, such as:
 - Internet (TCP/IP, mobile IP, DNS, DHCP)
 - ATM
 - CDPD
 - GSM
 - SNA
 - Wireless LANs
- Cryptographic protocols, such as:
 - Key distribution
 - Authentication
 - Payments
 - Anonymity and privacy
- Cryptographic functions, such as:
 - Encryption
 - Message authentication
 - Message digest
 - Signatures
- Computer viruses and worms
- Security for intelligent and mobile objects and agents
- Secure electronic commerce
- Cryptographic hardware
- Security and cryptography for wireless communication systems

Submissions E-mail submissions to one of the guest editors are encouraged (*Post-Script* only). Set your subject: field to "Submission to WINET special issue."

Guest Editors	Amir Herzberg IBM T.J. Watson Research Center P.O. Box 704 #H3-D18 Yorktown Heights, NY 10598 Phone: +1 914 784-6981 Fax: +1 914 784-6205 E-mail: amir@watson.ibm.com	Shay Kutten IBM T.J. Watson Research Center P.O. Box 704 #H3-D38 Yorktown Heights, NY 10598 +1 914 784-7346 +1 914 784-6205 kutten@watson.ibm.com
----------------------	---	---

Important dates	Manuscript Submission Deadline: November 15, 1995 Acceptance Notification: May 15, 1996 Final Manuscript Submission Deadline: July 15, 1996
------------------------	---

Call for Submissions

The *3rd International Workshop on Community Networking*, will be held May 23–24, 1996 (to be confirmed) in Antwerpen, Belgium. After the success of the first two CN workshops, held in San Francisco and in Princeton, the workshop now moves to Europe.

Objectives and Scope

New services such as Video-on-Demand, teleshopping, teleworking and edutainment are being introduced in communication networks. They will bring a variety of new applications to the user's home, and change the social and economic behavior of the—possibly virtual—communities they are living, working and communicating in. Experiences to date show that interactive service networks are primarily attractive to users, because they allow on-line communities to develop and flourish. The success of network technology, infrastructure and of the services deployed is therefore strongly determined by their potential for supporting communities. Continuously growing user access to and exchange of an overwhelming amount of information, provided by e.g., the World-Wide Web and other on-line services, not only introduces the need for appropriate network architectures and infrastructure, but also for creatively packaged content and well designed navigation tools. This in turn, stimulates the user's desire to actively participate in newly emerging electronic communities.

This workshop will give researchers and professionals from a variety of disciplines the opportunity to exchange their views and experiences, and advance the state of the art in the field. The 3rd edition's theme, "Stories behind the Picture," emphasizes the market's need for a clear vision on technology and services evolution, and care for real customer requirements. In this context, we encourage the submission of success stories on multimedia trials, Internet experiences, socially motivated experiments, and new technologies for interaction and communication.

Topics

Contributions are solicited on the following topics:

- *Network Architectures*: network topologies; addressing schemes; role of service gateways; intelligence in the network; billing systems; impact of access technologies; network evolution; network interworking; impact of standardization;
- *Services and Applications*: service creation; service management; service deployment; service brokerage; application platforms; STB operating systems and user interfaces; classification and parametrization; directory services; role of network operators; service interworking; service pricing and billing; authoring tools;
- *User Requirements and user interfaces*: navigation and interactivity; CPE; QoS networking requirements; regulation and legislation; community aspects; emerging services; service provider requirements;
- *Trials and Success Stories*: introduction strategies; social and economic impact; customer behavior; obstacles to success; business opportunities and threats; how networks can act as community builders;

Submissions

Contributions will be refereed from extended abstracts (in English), which should not be longer than 1,000 words. Send your abstract via e-mail (ASCII format) to: cn3@rc.bel.alcatel.be

Important dates

Extended abstracts due:	January 1st, 1996
Notification of acceptance:	February 15th, 1996
Camera-ready copy due:	April 1st, 1996

Book Reviews

Understanding Networked Multimedia, by François Flückiger (vice director of networking at CERN) published by Prentice Hall, 1995 (ISBN 0-13-190992-4), 620 pages.

Covers everything

This book covers everything about everything you need to know—and it is first rate on accuracy and timeliness. It is well written and goes from gentle introduction and principles right through to nitty gritty details (e.g., Mbone, ATM service models, MPEG, H.261, JPEG, G.711, conference control etc., etc., etc., brief intro to WWW, HTML, SGML, HyTime, MHEG etc., etc., etc.)

Organisation

The book is structured in 5 main sections, comprising 30 chapters in all. There is an excellent (welcome!) glossary of terms (40 pages), as well as an extensive bibliography. Throughout the book, important general questions are highlighted (as well as being forwarded referenced in a list at the start, and summarised in a list of key messages at the end), such as:

“Can I send TV over Ethernet?”

“What is an MCU?”

“Are ODA, MHEG or HyTime really different?”

“May an ATM Network be congested?”

...and so on.

and:

“Audio is significantly harder to compress than video.”

“The human ear behaves as a differentiator, the eye as an integrator.”

...and so forth.

The five sections are:

- *I Setting The Scene* (e.g., what is multimedia, who uses it, how and why?)
- *II Multimedia Applications* (audio/visual interpersonal, CSCW, Conferencing, Multimedia E-mail, Servers, Hypertext/WWW and VR).
- *III Network Requirements* (features, performance, etc.)
- *IV Network Solutions* (ATM, LANs, Packet and circuit WANs, Frame Relay and SMDS and futures, e.g., Mbone/RSVP etc.)
- *V Encoding and Compression* (basic techniques, principles, audio schemes such as G.721 etc, Video, including the JPEG, H.320 and MPEG series), and so on.

The book is written in a style that facilitates a straight read through (this reviewer read it on a long train journey), and moves from principles to practice and standards in each subsection, section and in theme and overall structure very elegantly.

Best text in the area

While it is early days to write a prescriptive text that might attempt to encompass all these ideas in some overall discipline or science, as the author so nicely notes in the introduction, there is plenty of scope for a book like this right now, and this is easily the best text in the area that this reviewer has seen. It even has a preface by Vint Cerf, as well as a former French minister for research!

Multimedia Programming by Simon J. Gibbs and Dionysios C. Tsichritzis, published by Addison-Wesley and the ACM Press, 1994, ISBN 0-201-42282-4, 323 pages. The authors are with the University of Geneva.

Prescriptive

The subtitle of this book is "Objects, Environments and Frameworks," and the authors, in their preface and acknowledgements, explain that their goal is in fact, quite prescriptive. As part of work at their University, and in the ESPRIT research program in Europe, they have started to capture the flavour of an architecture for multimedia systems.

Organisation

The book is in 3 parts, comprising 8 chapters:

Part 1 is about what multimedia comprises, and covers the media types, and multimedia programming environments. This contains an incredibly useful chapter (66 information packed pages) about CD, Phillips and Sony's CD-I, Intel's DVI, Apple's QuickTime and the Microsoft Multimedia PC standards (video for Windows and so on).

Part 2 describes the authors' own object oriented framework for comprehending multimedia system, and is quite an elegant, plausible framework along similar lines to that used in distributed systems by the ODP community.

Part 3 looks to the future, and covers some of the problems in multimedia systems, including composition/authoring, synchronisation, integration, and usability.

In 3 appendices, the authors describe more of their actual system, as well as listing useful information sources. The book is rounded off by a glossary and bibliography.

Highly useful

This book is highly useful. Most of the systems in use in the "multimedia industry" are in use for authoring material, though many commercial organisations are finding that the hype about multimedia is just that, and that existing authoring systems fall far short of what publishers (books, newspapers, TV and film industry) are accustomed too. From the material in this book, it is clear to see that many of the problems are really performance related, since the structure of the software systems the book describes seem well designed, and it can only be a matter of time before audio and video quality, storage capacity, and access speeds reach the point where the typical workplace PC is capable of real time composition and integration and playback of broadcast quality material. In the meantime, we can play with the prototypical systems that industry is tempting us with, and see what is right around the corner.

Comparison

To compare these two books would be a bit like comparing the pen and the wheel. The first book is about capturing, encoding, transmitting, decoding and receiving multimedia data, while the second is about the architecture of software systems to manipulate multimedia. They both belong on your bookshelf!

—Jon Crowcroft,
University College London
J.Crowcroft@cs.ucl.ac.uk

Letters to the Editor

Protocol Wars

Ole,

Your August 1995 issue features an article on Protocol Wars. I just wanted to say that as one of the minor participants in that "struggle," I believe a lot of credit for the TCP/IP victory should go to Dan Lynch. As most, if not all, of your readers know, in the mid '80s, Dan set up the Advanced Computing Environment's initiative to bring the TCP/IP research and development community together which eventually opened up the technology to the commercial world. The evolution of ACE into Interop broadened that effort and was literally overpowering. While attending one Interop conference some years ago, I ran into Vint Cerf. We both stood in the middle of the exhibition floor, marveling at what had been wrought. I am tickled pink by this victory, since I feel some personal vindication after being rebuffed by certain government organizations who rejected my continued suggestions that they formally adopt TCP/IP and forget the OSI suite.

Thanks for the article.

Regards,

—*Raphael (better known as Bob) Jones*
(raphaelj@edne.gov)

Bob,

On behalf of Dan Lynch and everyone involved in the evolution of Interop: thank you for your kind words. It's been inspiring to see the growth of a new industry through the twenty-two Interop events to date. Dan will officially retire from his role as chairman of Interop Company at the end of this year, but we are confident that we will continue to hear from him as he focuses on several Internet related projects such as CyberCash and The McKinley Group.

—*The Editor*

Spamming

Mr. Ole Jacobsen,

Attached is an item* that I think is a rather serious example of the spamming that "Michael Underwood" discusses in his article, "Street Sweeping The Information Super Highway" (*ConneXions*, Volume 9, No. 9). I can't imagine how I ended up on the receiving end of this item—the only mailing list I am aware of participating in being the RFC dist list...

And interestingly enough, when I tried to return a response to the sender, the address was invalid.

Seems like offensive spamming is pretty close to "hazmat"—and could be used against the recipient if someone were dreadfully vindictive.

I certainly support Mr. Underwood's suggestion, despite the difficult implementation issues that would have to be addressed!

—*Kimberly Hanson, Duke Power Company*
Charlotte, North Carolina

* The "attached item" has been removed to protect the guilty and the innocent. Suffice it to say that many readers would find it quite offensive.

—*The Editor*

More on Spamming

Ole,

In *ConneXions* 9.9 you have a paper from Michael Underwood about measures against spamming where he proposes a new feature for list management software that would require the sender to be a subscriber to the list.

Well, that feature has been part of the LISTSERV software as a configuration option for the last ten years or so. What is slightly more recent is that L-soft's LISTSERV is available for UNIX and not just for NJE, but even that is a few years now.

I enclose the address of an L-soft contact (well, the man who designed the software [Eric Thomas]) in case you or Michael want more information or if you want to give him advice on its development.

PS. I note that the old network boys are supposed to know Underwood's real identity. Well, I don't and I wonder if this is because I'm not a network boy or if I'm not old yet (?).

—Frode Greisen, UNI-C
Copenhagen, Denmark
(frode.greisen@uni-c.dk)

And Eric Thomas writes:

More importantly, LISTSERV automatically detects spams and forwards them to the list owner for human verification :—)

—Eric Thomas, SUNET
(ERIC@SEARN.SUNET.SE)

Write to *ConneXions*!

We'd love to hear your comments, suggestions and questions about anything you read in *ConneXions*. Our editorial address is given below. Use it for letters to the Editor, requests for the index of back issues, questions about particular articles etc.:

ConneXions—The Interoperability Report

303 Vintage Park Drive

Foster City

California 94404-1138

USA

Phone: +1 415-578-6900 or 1-800-INTEROP (Toll-free in the USA)

Fax: +1 415-525-0194

E-mail: connexions@interop.com

URL: <http://www.interop.com>

Subscription information

For questions about your subscription please call our customer service hotline: 1-800-575-5717 or +1 610-892-1959 outside the USA. This is the number for our new subscription agency, Seybold Publications. Their fax number is +1 610-565-1858. The mailing address for subscription payments is: P.O. Box 976, Media, PA 19063-0976.

This publication is distributed on an "as is" basis, without warranty. Neither the publisher nor any contributor shall have any liability to any person or entity with respect to any liability, loss, or damage caused or alleged to be caused, directly or indirectly, by the information contained in *ConneXions—The Interoperability Report*®

CONNEXIONS

303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

ADDRESS CORRECTION
REQUESTED

CONNEXIONS

EDITOR and PUBLISHER Ole J. Jacobsen

EDITORIAL ADVISORY BOARD Dr. Vinton G. Cerf
Senior Vice President, MCI Telecommunications
President, The Internet Society (1992 – 1995)

A. Lyman Chapin, Chief Network Architect,
BBN Communications

Dr. David D. Clark, Senior Research Scientist,
Massachusetts Institute of Technology

Dr. David L. Mills, Professor,
University of Delaware

Dr. Jonathan B. Postel, Communications Division Director,
University of Southern California, Information Sciences Institute



Printed on recycled paper

Subscribe to CONNEXIONS

U.S./Canada ☐ \$150. for 12 issues/year ☐ \$270. for 24 issues/two years ☐ \$360. for 36 issues/three years

International \$ 50. additional per year (Please apply to all of the above.)

Name _____ Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Telephone () _____

☐ Check enclosed (in U.S. dollars made payable to CONNEXIONS).

☐ Visa ☐ MasterCard ☐ American Express ☐ Diners Club Card # _____ Exp. Date _____

Signature _____

Please return this application with payment to:

CONNEXIONS

303 Vintage Park Drive, Suite 201

Foster City, CA 94404-1138

415-578-6900 FAX: 415-525-0194

connexions@interop.com

Back issues available upon request \$15./each
Volume discounts available upon request

CONNEXIONS